

## 008AA – ALGORITMICA E LABORATORIO

Appello del 14 luglio 2010

### Esercizio 1.

Scrivere un algoritmo *divide et impera* che, dato un array  $a$  **ordinato** di interi **distinti** (anche negativi), verifichi se esiste un indice  $i$  tale che  $a[i] = i$ . Analizzare la complessità dell'algoritmo proposto, indicando e risolvendo la corrispondente relazione di ricorrenza.

### Soluzione

```
TrovaPosizione(a, sx, dx)
    if (sx > dx) return false;
    if (sx == dx) return (a[sx] == sx);
    centro = (sx + dx) / 2;
    if (a[centro] == centro) return true;
    if (a[centro] > centro) return TrovaPosizione(a, sx, centro-1);
    else return TrovaPosizione(a, centro+1, dx);
```

### Costo in tempo:

$$T(n) = T(n/2) + O(1),$$

Teorema Principale, secondo caso:  $T(n) = O(\log n)$

### Esercizio 2.

Date le stringhe  $A = CIAMBELLA$  e  $B = MENSOLA$ , simulare l'algoritmo di programmazione dinamica per il calcolo della lunghezza della sottosequenza comune più lunga, mostrando

1. il contenuto della tabella che l'algoritmo riempie dinamicamente;
2. la soluzione trovata dall'algoritmo.

### Soluzione

	M	E	N	S	O	L	A	\$
C	4	3	2	2	2	2	1	0
I	4	3	2	2	2	2	1	0
A	4	3	2	2	2	2	1	0
M	4	3	2	2	2	2	1	0
B	3	3	2	2	2	2	1	0
E	3	3	2	2	2	2	1	0
L	2	2	2	2	2	2	1	0
L	2	2	2	2	2	2	1	0
A	1	1	1	1	1	1	1	0
\$	0	0	0	0	0	0	0	0

La soluzione è la stringa  $MELA$ , di lunghezza 4.

### Esercizio 3.

Descrivere lo pseudocodice di una visita in profondità di un grafo generico  $G$ , e calcolarne la complessità.

### Soluzione

Si veda il libro di testo.

### Esercizio 4.

Si scriva una funzione `isPalindroma(char *s)` in linguaggio C che, data una stringa di caratteri priva di separatori, restituisce 1 se tale parola è *palindroma* (cioè leggibile nello stesso modo da destra a sinistra e viceversa, come OSSESSO) e 0 altrimenti.

Si scriva anche la funzione `main()` che chiede all'utente di inserire una stringa da tastiera e testi `isPalindroma()`. Nel `main()` si utilizzi la funzione `malloc()` per allocare lo spazio necessario alla memorizzazione della stringa.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char *s;
    s=malloc(100*sizeof(char));
    printf("Inserisci stringa: ");
    scanf ("%s",s);
    printf("Stringa inserita: %s",s);
    printf("Stringa Palindroma? %d\n",isPalindromo(s));
}

int isPalindromo(char *s){

    int flag = 1;
    int i=0;
    while ((i < (strlen(s)/2)) && (flag == 1)) {
        if (s[i] != s[strlen(s)-i-1]) {
            flag = 0;
        }
        i++;
    }
    if (flag == 1) {
        return 1;
    } else {
        return 0;
    }
}
```